Network subscriber station for a network of distributed stations, and method for operating a network subscriber station

5   The invention relates to a network subscriber station
     for a network of distributed stations, particularly in
     connection with IEEE 1394 network subscriber stations,
     and to a method for operating a network subscriber
     station.

10

Prior art

     Interfaces   are   used   in   connection   with
     electrical/electronic appliances in order to allow the
     individual   appliance   to   communicate   electronically,
15   particularly to send and receive electronic data, which
     are   usually   transmitted   using   a   data   bus   which   is
     connected to the appliance. Depending on the data bus
     used   and   on   the   electrical/electronic   appliances
     connected to the data bus, there are various standards
20   for interface configurations. One interface standard to
     which   a   great   deal   of   attention   has   been   devoted
     recently is the IEEE 1394 standard. IEEE 1394 provides
     a standard which is particularly suitable for assisting
     data   interchange   between   multimedia   devices   of   any
25   kind. A data bus which is configured on the basis of
     the   IEEE 1394   standard   can   be   connected   both   to
     peripheral devices associated with a computer, such as
     printers,   scanners,   CD drives   and   hard   disks,   and   to
     domestic   electronic   appliances,   such   as   video   cameras
30   or   televisions.   A   great   deal   of   influence   on   the
     digitization   of   electronic   appliances   is   therefore
     expected from this standard.

     The IEEE 1394 standard involves a reset operation ("bus
35   reset")   being   carried   out   on   the   data   bus   when   an
     electronic   appliance   is   connected   to   the   data   bus,   an
     electronic   appliance   is   disconnected   from   the   data   bus
     or   when   the   configuration   data   for   an   appliance   have

- 2 -

changed. In the case of a reset operation on the data
bus, every IEEE 1394 interface node which is connected
to the data bus sends an independent ID
(identification) information item to the other network
subscriber stations. This ensures that every network
subscriber station in a network of distributed stations
connected by means of the data bus is informed about
which other network subscriber stations are connected
in the network. The independent ID information item is
used by each interface node to identify itself to the
other interface nodes in the network. Using the
independent ID information item received from the other
respective interface nodes, each interface node in the
network is able to create an "interface node list" and
to store it in a memory device respectively associated
with the interface node. This stored information item
can then be processed by a driver program on the
respective network subscriber station when managing the
data packet to be transmitted and received via the data
bus.

In addition, a reset operation on the data bus serves
as a request to the interface nodes in the connected
electrical/electronic appliances to read each other's
interface configuration data. The interface
configuration data are stored in memory areas in a ROM
(Read Only Memory) store in the network subscriber
station, which are also referred to as ConfigROM memory
areas. The content of the interface configuration data
in the memory areas is operation-dependent, i.e. it can
change during operation of the network. Such a change
may result in the stored interface configuration data
being changed while a reset operation and the
associated request for reciprocal reading of the
interface configuration data is being received. This
results in the problem that sometimes inconsistent
and/or incomplete interface configuration data are

- 3 -

read, since the interface configuration data are edited at a parallel time on account of changes on the respective electrical/electronic appliance.

5 To avoid this problem, known chipsets based on the "1394 Open Host Controller Interface, Release 1.1" specification already comprise options for measures. In this context, provision is made for a register which contains an electronic pointer to a current ROM memory
10 area to be accompanied by the implementation of a further electronic pointer which points to a further ROM memory area which stores interface configuration data which are provided for reading after a subsequent reset operation on the data bus. The further electronic
15 pointer is not adopted as the new address for the current ROM memory area during a subsequent reset operation until a read request which has been started has been executed. However, this means that the reading of inconsistent interface configuration data cannot be
20 prevented. If, by way of example, a reset operation occurs while the interface configuration data are being edited in the further ROM memory area, then there is the risk that incompletely changed interface configuration data will be read.
25

The appliance reading the interface configuration data can – if it establishes that there is an inconsistency – repeat the read operation, e.g. can take place in connection with a preceding further reset operation.
30 However, this results in increased data traffic on the data bus and hence in increased reaction times in the network of network subscriber stations. In most cases, the inconsistency in the interface configuration data is not noticed at first, however, since many appliances
35 carry out no or only inadequate CRC (Cyclic Redundancy Check, check sum evaluation) checks on account of the increased reading involvement associated therewith. A

- 4 -

data record which is output incorrectly in this manner can result in serious disruptions to the way in which the network works.

5   Invention
It is an object of the invention to specify an improved network subscriber station and also an improved method for operating the network subscriber station in a network of distributed stations where any operating
10   situations when reading ROM memory areas with interface configuration data have the assurance of reading being independent of possible operations for editing the interface configuration data.

15   This object is achieved by a network subscriber station based on the independent claim 1 and a method for operating the network subscriber station based on the independent claim 3.

20   The invention covers the concept of providing at least three memory areas for operation-dependent interface configuration data in the implemented data link layer in the case of a network subscriber station for a network of distributed stations, particularly a network
25   of IEEE 1394 network subscriber stations, which are connected by means of a data bus and have a respective implemented data link layer. Pointer means, which comprise electronic pointers to the at least three memory areas, can be used to access the at least three
30   memory areas, which means that stored data can be read and/or written in. In addition, driver means are provided in order to handle electronic data in the at least three memory areas and to transfer them between the at least three memory areas.
35

Using the at least three memory areas, it is possible to use a plurality of memory areas to store interface

- 5 -

configuration data such that a current, complete and inherently consistent set of interface configuration data is always available for reading the interface configuration data on account of a reset operation on

5 the data bus. To achieve this, various versions of the interface configuration data can be transferred arbitrarily between the plurality of memory areas according to a scheme which is possibly dependent on the implementation. Correct readable interface

10 configuration data are available at all times. There is no increased data traffic which is required on account of repetitions of incorrectly handled read requests after a reset operation in connection with known methods. As a result, the reaction time and hence the

15 operating convenience of the network with the electrical/electronic appliances, which are coupled via interface nodes, are also improved.

Drawing

20 The invention is explained in more detail below using an exemplary embodiment with reference to a drawing, in which:

figure 1   shows a schematic illustration of a network
25   with electrical/electronic appliances which
               are connected to a data bus via a respective
               interface node;

figure 2   shows a schematic flowchart for an edit
30   operation for interface configuration data;
               and

figure 3   shows an overview of address pointer use in
               table form.

35

Exemplary embodiment of the invention
Figure 1 shows a schematic illustration of a network in

- 6 -

which a plurality of network subscriber stations 2 are
connected to a data bus 1 via a respective interface
node 3 for the purpose of sending/receiving data
packets via the data bus 1. The respective interface
node 3 is in the form of an IEEE 1394 interface node,
which allows, in particular, coupling/decoupling of the
network subscriber stations 2 and data packet
interchange between the network subscriber stations 2
connected to the data bus 1. The interface device
comprises hardware means 4 and software means 5 for
operating the hardware means 4. The hardware means 4
are used, in particular, to send and receive data
packets via the data bus 1. The software means 5
comprise, in particular, a driver program for operating
the hardware means 4, in order to provide the
functionalities of the network subscriber station 2. A
piece of application software 6 respectively installed
in the network subscriber station 2 uses the interface
node 3 in order to send or receive data packets via the
data bus 1.

Particulars relating to the design and configuration of
an interface based on the IEEE 1394 standard are known
to the person skilled in the art and are therefore not
explained in detail at this point. Thus, an IEEE 1394
interface normally has a three-layer structure
comprising a transaction layer, a data link layer and a
physical layer. A bus management function monitors and
controls the three layers and provides various
functions for controlling the respective interface
node. In this context, the transaction layer provides
an interface (a service) for transactions with layers
arranged above the interface node. The data link layer
essentially provides the following functions:
addressing, data checking, data formation for packet
transmission/packet reception etc. The physical layer
converts logical symbols used by the data link layer

- 7 -

into electrical signals, performs data bus equalization and defines the physical data bus interface. The physical layer and the data link layer are normally produced using hardware means, for example as a data
5   transfer control device (interface chip). Separate ICs for the two layers are available on the market. The transaction layer is normally produced using firmware, which runs on a processor, or hardware.

10 With reference to figures 2 and 3, the text below gives a detailed description of the sequence for storing and handling the interface configuration data in a plurality of ROM memory areas. The ROM memory areas do not need to be in the form of physical ROM stores.
15 Instead, these memory areas may have been created in the RAM store in the respective network station. This is done by a memory management unit in the network station. The memory management unit reserves/allocates the necessary memory areas. The RAM store is not
20 exclusively associated with the data link layer. Other components may also access it, particularly a microcontroller in the network station which is controlled by the software means for implementing the higher layers.
25

In this context, figure 2 schematically shows a flowchart for method steps during an edit operation for interface configuration data. Before editing, an electronic address pointer is set, in the course of bus
30 reset handling, as follows: ActualPtr = NextPtr. As figure 2 shows, the edit operation then comprises a total of eleven instructions (0) ... (10), which are shown in figure 2 by means of pseudocode.

35 The instructions (0) and (10) (Lock() and Unlock()) are necessary in order to prevent simultaneous edit operations by a plurality of applications.

- 8 -

In the description below, it is assumed that the
network subscriber stations have three ROM memory areas
implemented in them for which the labels ConfigROM A,
5   ConfigROM B and ConfigROM C are chosen. The ROM memory
areas can be accessed by the driver means in the
associated network subscriber station using electronic
address pointers ActualPtr, NextPtr and EditPtr.
Depending on the method level, the electronic address
10   pointers ActualPtr, NextPtr and EditPtr can be loaded
into three address registers. ActualPtr points to the
memory area in the ConfigROM which contains the
interface configuration data for a current read
request. NextPtr points to the memory area in the
15   ConfigROM which contains changed interface
configuration data for answering read requests after a
subsequent reset operation. EditPtr points to the
memory area in the ConfigROM in which changes can be
made to the interface configuration data stored
20   therein. In addition, two buffer areas AuxPtr and
TempPtr for address pointers are provided which as
auxiliary memory areas are likewise implemented in the
memory area.

25   Figure 3 shows a tabular overview of the effect of
changes in the interface configuration data stored in
the ConfigROMs A, B, C and the effect of reset
operations on the contents of the address registers,
and also the respective (present) operation of the
30   ConfigROMs A, B, C during the sequence described. In
the tabular overview shown in figure 3, the following
five scenarios are described: T0-T5: editing with
subsequent reset operation; T10-T19: editing and re-
editing of the interface configuration data and
35   subsequent reset operation on the data bus; T30-T33:
reset operation after the end of the second edit
between instructions (0) and (1) (cf. figure 2); T40-

T43: reset operation after the end of the second edit between the instructions (4) and (5); and T50-T54: reset operation after the end of the second edit between the instructions (5) and (6).

5

The time T0 represents an initial state in which the ConfigROM A contains a data record 0 and is used for current read requests as a result of reset operations on the data bus. NextPtr likewise points to the

10 ConfigROM A, since no other changed interface configuration data for use after a subsequent reset operation exist. ConfigROM B, which likewise contains the data record 0, can hold changes. At the time T1, ConfigROM B is changed. In the course of the change,

15 the data record 0 containing interface configuration data is changed to the data record 0' containing changed interface configuration data. The editing carried out in the process has finished at time T2. ConfigROM B then contains a data record 1 containing

20 changed interface configuration data for which the change is complete.

At time T3, address pointer changeover takes place. When carrying out the instructions (1) ... (4) (cf.

25 figure 2), the operation (ActualPtr == EditPtr) gives the result "FALSE", which is why the instructions (6) ... (8) are not executed. Next, at time T4, a copy operation takes place, which involves the data content of the memory area to which the NextPtr (B(1)) points

30 being copied to the area to which the EditPtr points. At time T5, a reset operation now occurs on the data bus and prompts reading of the interface configuration data. During the reset operation, the ActualPtr is set to the value of NextPtr. All read requests which now

35 occur as the result of a reset operation at the interface nodes are handled using the data record from ConfigROM B (data record 1).

- 10 -

At time T10, there follows a further editing phase, which is finished at time T11. At time T12, address pointer changeover again takes place, which leads to

5   the result "FALSE" being obtained after execution of the instructions (1) ... (4) (cf. figure 2). This is followed by another copy operation, further editing and termination of the further editing at times T13, T14, T15. At time T16, address pointer changeover takes

10   place again. Unlike in the case of previous executions of the instructions (1) ... (4), "TRUE" is now obtained, which means that the instructions (6) ... (8) are executed at time T17. As a result, the entries in EditPtr and AuxPtr are interchanged. The effect

15   achieved by this is that EditPtr points to ConfigROM C. ConfigROM C is available for editing after the subsequent reset operation on the data bus. Next, the current data record is copied to ConfigROM C at time T18. As a result of a reset operation at time T19,

20   ActualPtr now points to the newest data record.

The scenarios which follow in the tabular overview in figure 3 begin, as a succession of two edit operations which have already been performed without an interim

25   reset operation, with three different data records from interface configuration data in the ConfigROMs A, B, C (respective starting points T30, T40, T50). The tabular overview in figure 3 shows that a reset operation may arise at any time in the sequence without edit

30   operations endangering the consistency of the current ConfigROM.

The use of the three ConfigROMs A, B, C in the manner described prevents the occurrence and transmission of

35   inconsistent interface configuration data as a result of a reset operation on the data bus, even if reset operations occur during the change operation for

- 11 -

editing the interface configuration data, since only
ever the interface configuration data in the memory
area in the ConfigROMs are edited which is used neither
at the current time nor after the next reset operation
5   for current enquiries, that is to say a temporary
inconsistency might exist. Hence, there is always a ROM
memory area available for interface configuration data
to be changed which is independent of a current and a
subsequent read operation, which are respectively
10   triggered as a response to a reset operation.